History Matching Through a Smooth Formulation of Multiple-Point Statistics

Melnikova, Yulia; Zunino, Andrea; Lange, Katrine; Cordua, Knud Skou; Mosegaard, Klaus

Published in:
Mathematical Geosciences

Link to article, DOI:
10.1007/s11004-014-9537-y

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Math Geosci (2015) 47:397–416
DOI 10.1007/s11004-014-9537-y

History Matching Through a Smooth Formulation of Multiple-Point Statistics

Yulia Melnikova · Andrea Zunino · Katrine Lange · Knud Skou Cordua · Klaus Mosegaard

Received: 12 August 2013 / Accepted: 11 April 2014 / Published online: 20 May 2014

Abstract We propose a smooth formulation of multiple-point statistics that enables us to solve inverse problems using gradient-based optimization techniques. We introduce a differentiable function that quantifies the mismatch between multiple-point statistics of a training image and of a given model. We show that, by minimizing this function, any continuous image can be gradually transformed into an image that honors the multiple-point statistics of the discrete training image. The solution to an inverse problem is then found by minimizing the sum of two mismatches: the mismatch with data and the mismatch with multiple-point statistics. As a result, in the framework of the Bayesian approach, such a solution belongs to a high posterior region. The methodology, while applicable to any inverse problem with a training-image-based prior, is especially beneficial for problems which require expensive forward simulations, as, for instance, history matching. We demonstrate the applicability of the method on a two-dimensional history matching problem. Starting from different initial models we obtain an ensemble of solutions fitting the data and prior information defined by the training image. At the end we propose a closed form expression for calculating the prior probabilities using the theory of multinomial distributions, that allows us to rank the history-matched models in accordance with their relative posterior probabilities.

Keywords History matching · Multiple-point statistics · Optimization · Inverse problems

Y. Melnikova (✉) · A. Zunino · K. S. Cordua · K. Mosegaard
Division of Mathematical and Computational Geoscience,
Center for Energy Resources Engineering, National Space Institute, Technical University of Denmark,
Richard Petersens Plads Building 324, 2800 Kongens Lyngby, Denmark
e-mail: yume@dtu.dk

K. Lange
Banedanmark, Copenhagen, Denmark

1 Introduction

History matching is a task of inferring knowledge about subsurface models of oil reservoirs from production data. History matching is a strongly underdetermined problem: having data in a limited number of wells, one needs to estimate rock properties in the whole reservoir model. This problem has infinitely many solutions, and in addition, most of them are not geologically plausible. Furthermore, the intensive computational work needed to simulate the data redoubles the complexity. To address these challenges, we develop a probabilistic framework that combines complex a priori information and simultaneously aims at reducing the number of forward simulations needed for finding solutions. We propose a smooth formulation of the inverse problem with discrete-facies prior defined by a multiple-point statistics model. This allows us to use gradient-based optimization methods to search for feasible models. In probabilistic inverse problem theory (Tarantola 2005) the solution of an inverse problem is represented by its a posteriori probability density function (PDF). Each possible state in the model space is assigned a number—a posteriori probability density—which reflects how well the model honors the data and the a priori information (knowledge about the model parameters independent from the data). The a posteriori PDF of high-dimensional, underdetermined inverse problems, such as history matching, may feature isolated islands of significant probabilities and low probabilities everywhere else. Therefore, when the full description of the posterior PDF is not available, the goal is to locate and explore islands of significant posterior probabilities.

One may explore the a posteriori PDF in several ways. Monte Carlo methods (Mosegaard and Tarantola 1995; Cordua et al. 2012) allow, in principle, sampling of the a posteriori PDF. However, for large scale non-linear inverse problems, there is a risk of detecting only a single island of significant posterior probability. In addition, sampling is not feasible for inverse problems with computationally expensive forward simulations, such as history matching. Other methods rely on optimization (Caers and Hoffman 2006; Jafarpour and Khodabakhshi 2011) to determine a collection of models that fit the data and the a priori information. However, these methods fail to describe a posteriori variability of the models as the weighting of prior information versus data information (likelihood) is not taken into account.

Regardless of the chosen strategy, most of the research community favors the advanced prior information that helps to significantly shrink the solution space of allowed models (Caers 2003; Jafarpour and Khodabakhshi 2011; Hansen et al. 2012). For instance, the a priori information borrowed from a training image (Guardiano and Srivastava 1993; Strebelle 2002) would permit only models of a specific configuration defined by statistical properties of the image. Ideally, training images reflect expert knowledge about geological phenomena (facies geometry, contrast in rock properties, location of faults) and play a role of vital additional information, drastically restricting the solution space (Hansen et al. 2009). Our strategy for exploring the a posteriori PDF, which is especially suitable for inverse problems with expensive forward simulation (e.g. history matching), is to obtain a set of models that feature high posterior values, and rank the solutions afterwards in accordance with their relative posterior probabilities. We integrate complex a priori information represented by multiple-point statistics inferred from a training image. One of the challenges here is to define a closed form

expression for the prior probability that, multiplied by the likelihood function, provides the a posteriori probability. It is not sufficient to perturb the model in consistency with the training image until the dynamic data are matched as it is done in the probability perturbation method (Caers and Hoffman 2006). As it was noticed by Hansen et al. (2012), in this method the fit to the prior information is not quantified, so the method will spot models of maximum likelihood/non-zero prior, not of maximum posterior; the resulting model may resemble the training image very poorly, and therefore may have a low posterior value.

Lange et al. (2012) were the first who aimed at estimating prior probabilities solving inverse problems with training images. The developed frequency matching (FM) method is able to quantify the prior probability of a proposed model and hence to iteratively guide it towards the high posterior solution. Specifically, Lange et al. (2012) solve a combinatorial optimization problem, perturbing the model in a discrete manner until it explains both data and a priori information. In practice, this requires many forward simulations and can be prohibitive for the history matching problem. While following the philosophy of the frequency matching method, we are interested in minimizing the number of forward simulations needed to achieve a model of a high posterior probability. Similarly to the FM method, we minimize the sum of data and prior misfits. However, the new smooth formulation of the objective function allows us to apply gradient-based optimization and sufficiently cut down the number of reservoir simulations. After convergence the model has all statistical properties of the training image and simultaneously fits the data. Having several starting models, possibly very different, we are able to obtain different solutions of the inverse problem and to detect regions of high posterior probability. In the case of the history matching problem, starting models obtained from seismic data interpretation probably would be of most practical use.

To our knowledge, gradient-based techniques were first coupled with training images in the work of Sarma et al. (2008) by means of kernel principal component analysis (PCA). The authors were the first who used kernel PCA for geological model parametrization. The kernel PCA generates differentiable (smooth) realizations of the training image, maintaining its multiple-point statistics and, as a result, reproducing geological structures. The differentiable formulation by Sarma et al. (2008) allows the use of gradient-based methods; however, the quality of the solution in terms of consistency with the prior information is not estimated. In this work, we actually derive a closed form expression for the prior probability. This allows us to quantify the relative posterior probabilities of the solutions and therefore to assess their importance.

This paper is organized as follows. In Sect. 2, we introduce the smooth formulation of multiple-point statistics. The proposed formulation makes it possible to measure the mismatch between multiple-point statistics of the training image and of any, possibly continuous, model. As the result, we are able to generate realizations of the training image from any starting model image using gradient-based optimization (Sect. 2.4). Combination of the proposed measure with the data misfit allows us then to search a solution to an inverse problem with training-image-based prior by minimizing a single differentiable objective function (Sect. 2.5). In Sect. 3, we demonstrate the applicability of the method solving a two-dimensional history matching problem. At the end, we rank the solutions in accordance with their relative posterior probabilities using derivations from Sect. 2.3. Section 4 summarizes our findings.

2 Methodology

In this work, we use a probabilistic formulation of inverse problems, integrating complex a priori information (training image) and data into a single differentiable objective function. Solving the optimization problem for an ensemble of starting models we obtain a set of solutions that honor both the observations and multiple-point statistics of the training image. We start with a definition of the inverse problem.

2.1 Inverse Problems with Training Image-Defined Prior

Denoting the model parameters as $\mathbf{m}$, the non-linear forward operator as g and its response as $\mathbf{d}$, we introduce the forward problem

$$\mathbf{d} = g(\mathbf{m}). \tag{1}$$

The inverse problem is defined then as the task of inferring the model parameters $\mathbf{m}$ given the observed data $\mathbf{d}^{\mathbf{obs}}$, the forward relation g and, if available, some (data independent) a priori information about model parameters. Addressing inverse problems, we employ a probabilistic approach (Tarantola 2005), where the solution is characterized by its a posteriori PDF. The a posteriori PDF $\sigma(\mathbf{m})$ contains the combined information about the model parameters as provided by the a priori PDF $\rho(\mathbf{m})$ and the likelihood function $L(\mathbf{m})$

$$\sigma(\mathbf{m}) = k\ \rho(\mathbf{m}) L(\mathbf{m}), \tag{2}$$

where k is a normalization constant. The likelihood function $L(\mathbf{m})$ measures how well the model $\mathbf{m}$ fits the observations $\mathbf{d}^{\mathbf{obs}}$

$$L(\mathbf{m}) = c_1 \exp\left(-\frac{1}{2}||g(\mathbf{m}) - \mathbf{d}^{\mathbf{obs}}||^2_{C_D}\right), \tag{3}$$

where c_1 is a constant and C_D is the covariance matrix representing Gaussian uncertainties in the measurements. Prior information is assumed to be obtained from a training image with discrete pixel (voxel) values, representing some subsurface property. In this case, the expression for the a priori probability density function is known explicitly (Lange et al. 2012)

$$\rho(\mathbf{m}) = c_2 \exp(-\alpha f(\mathbf{m}, \mathbf{TI})), \tag{4}$$

where the function $f(\mathbf{m}, \mathbf{TI})$ measures the dissimilarity between the multiple-point statistics of the training image $\mathbf{TI}$ and the model $\mathbf{m}$; c_2 is a normalization constant, α is the problem-dependent weight factor. The statistics has the form of the frequency distribution of the observed patterns in the image. A pattern is a set of neighboring pixels in the image of shape defined by a template $\mathbf{T}$. Consider, for instance, a 2×2

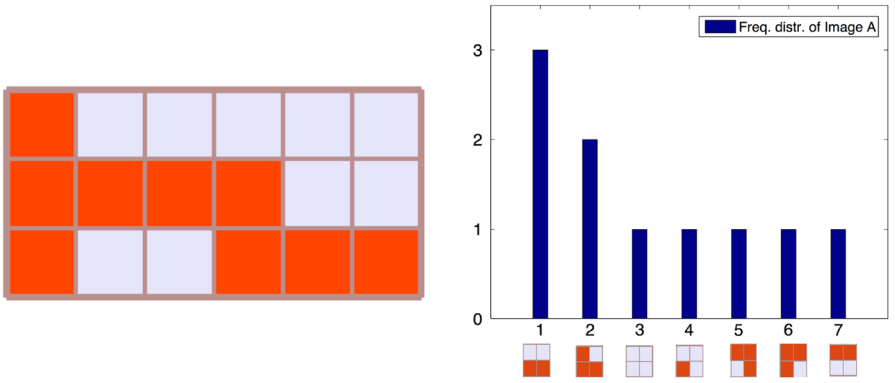

Fig. 1 Discrete image A (**a**) and its pattern frequency distribution (**b**); 2×2 template applied

square template applied to the binary image shown in Fig. 1a and the obtained histogram shown in Fig. 1b (only non-zero counts out of possible 16 combinations are shown).

Constructing such histograms for the training and the model images, Lange et al. (2012) define their statistical dissimilarity $f(\mathbf{m}, \mathbf{TI})$ by calculating the chi-square distance between the histograms. The closed form expression for the a priori PDF (Eq. 4) enables us to estimate the value of the posterior probability of a given model as well as to search for a maximum a posteriori solution. Lange et al. (2012) find the maximum a posteriori solution of the inverse problem minimizing the following sum of misfits

$$\mathbf{m}^{\mathrm{MAP}} = \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \frac{1}{2} ||\mathbf{d}^{\mathbf{obs}} - g(\mathbf{m})||^2_{C_D} + \alpha f(\mathbf{m}, \mathbf{TI}) \right\}. \tag{5}$$

The FM method defines the a priori PDF as a function of frequency distributions of the patterns, not of the pixel values. This leads to two limitations: the prior probability can be estimated only for discrete images, whose categorical values are identical to those of the training image; in optimization the model image should stay discrete. In other words, Eq. 5 is a combinatorial optimization problem that typically requires running a large number of forward simulations. Lange et al. (2012), for instance, used the simulated annealing algorithm which required several thousands of forward runs to achieve the solution. Aiming at minimizing the number of forward simulations (flow simulations) we suggest an alternative approach, which is based on a smooth formulation of multiple-point statistics. The smooth formulation (Sect. 2.2) allows us to solve an optimization problem similar to Eq. 5 using gradient-based optimization.

The goal is to gradually change a starting model $\mathbf{m}$ into a model $\mathbf{m}^{\mathrm{HighPosterior}}$ with high posterior value, that is into one that honors both data and prior information. To this end, we introduce a differentiable function $f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI})$ which measures the mismatch between the multiple-point statistics of the training image and the model. We show how by minimizing the value of the proposed measure we are able to generate images

that honor multiple-point statistics of the training image. Finally, a solution to the inverse problem is found by solving the following optimization problem

$$\mathbf{m}^{\text{HighPosterior}} = \underset{\mathbf{m}}{\text{argmin}} \left\{ \frac{1}{2} ||\mathbf{d}^{\mathbf{obs}} - g(\mathbf{m})||^2_{C_D} + f^d(\mathbf{m}, \mathbf{TI}) \right\}. \tag{6}$$

Notice the absence of the weight factor α in comparison with Eq. 5.

2.2 The Smooth Formulation of Multiple-Point Statistics

In this section, we derive a differentiable function $f^{\text{d}}(\mathbf{m}, \mathbf{TI})$ that allows us to measure dissimilarity between the multiple-point statistics of the discrete training image and of any continuous image. To this end, we introduce a new object called pseudo-histogram (or smooth histogram), which reflects pattern statistics of an image. In contrast to the frequency distribution, it is a function of pixel values, not of the pattern counts, and it can be computed for both discrete and continuous images. It has an important property: for the training image it almost coincides with its frequency distribution. We then compare training and model images by comparing their pseudo-histograms, which are differentiable with respect to model parameters. For clarity we use two-dimensional images in our explanation, though the algorithm is implemented for both two- and three-dimensional problems. Our notation is presented in Table 1.

Assume that the prior information is represented by a categorical training image **TI**, whose pixel values are real numbers (e.g. 10.0 and 500.0) and represent some physical property (e.g. permeability). First, we scan through the training image **TI** using the template **T** and save its unique (non-repeating) patterns as a database. Unique patterns of the training image define categories of the pseudo-histograms $H^{\text{d},\mathbf{m}}$ and $H^{\text{d},\mathbf{TI}}$. We show in detail how to construct the pseudo-histogram for the model image only, noticing that $H^{\text{d},\mathbf{TI}}$ is constructed in the same manner. The approach is based on the idea that a continuous pattern $\text{pat}_i^{\mathbf{m}}$ observed in the image **m** does not fit into a single discrete pattern category $\text{pat}_j^{\mathbf{TI},\text{un}}$, but instead it contributes to all $N^{\mathbf{TI},\text{un}}$ categories.

Table 1 Notation

Notation	Description
TI	Training image, categorical
m	Model (test image), can contain continuous values
T	Scanning template
$H^{\text{d},\mathbf{m}}$	Pseudo-histogram of **m**
$H^{\text{d},\mathbf{TI}}$	Pseudo-histogram of **TI**
$N^{\mathbf{m}}$	Number of patterns in **m**
$N^{\mathbf{TI}}$	Number of patterns in **TI**
$N^{\mathbf{TI},\text{un}}$	Number of unique patterns in **TI**
$\text{pat}_i^{\mathbf{m}}$	Pixel values of ith pattern in **m**
$\text{pat}_i^{\mathbf{TI}}$	Pixel values of ith pattern in **TI**
$\text{pat}_j^{\mathbf{TI},\text{un}}$	jth unique pattern in **TI**.

Therefore, summing over all the $N^{\mathbf{m}}$ contributions, the jth bin of the pseudo-histogram $H^{\mathrm{d},\mathbf{m}}$ is defined as

$$H_j^{\mathrm{d},\mathbf{m}} = \sum_{i=1}^{N^{\mathbf{m}}} c_{ij}, \tag{7}$$

where c_{ij} defines the level of similarity between $\mathrm{pat}_i^{\mathbf{m}}$ and $\mathrm{pat}_j^{\mathbf{TI},\mathrm{un}}$. We define c_{ij} such that it equals 1 when vector of pixel values $\mathrm{pat}_i^{\mathbf{m}}$ is equal to $\mathrm{pat}_j^{\mathbf{TI},\mathrm{un}}$. A natural choice for c_{ij} would be one based on the Euclidean distance between pixel values of the corresponding patterns, defined, for instance, as

$$c_{ij} = \frac{1}{\left(1 + A\, t_{ij}^k\right)^s}, \tag{8}$$

where $t_{ij} = ||\mathrm{pat}_i^{\mathbf{m}} - \mathrm{pat}_j^{\mathbf{TI},\mathrm{un}}||_2$ and A, k, and s are the user-defined parameters (scalars).

Notice the following property

$$c_{ij} = \begin{cases} 1 & t_{ij} = 0 \\ \in (0,1) & t_{ij} \neq 0. \end{cases} \tag{9}$$

In the same manner, we define the smooth histogram for the training image itself

$$H_j^{\mathrm{d},\mathbf{TI}} = \sum_{i=1}^{N^{\mathbf{TI}}} \frac{1}{\left(1 + A\, ||\mathrm{pat}_i^{\mathbf{TI}} - \mathrm{pat}_j^{\mathbf{TI},\mathrm{un}}||_2^k\right)^s}. \tag{10}$$

The smooth histogram computed for the discrete Image A (Fig. 2a is shown in Fig. 2c by light-blue color, while its original frequency distribution is depicted by the dark-blue color. Categories of discrete patterns, contributions to which are calculated using Eq. 8, are shown below the x-axis. Figure 2b shows a continuous image, while in Fig. 2c one can see its histogram, defined in the smooth sense, depicted by the orange color. Notice the small counts everywhere: indeed, according to Eq. 9, this image does not contain patterns sufficiently similar to those observed in the training image. For the visualization purposes parameters of Eq. 8 are chosen as $A = 50, k = 2$ and $s = 2$. These values are applicable after t_{ij} has been normalized on the quantity representing maximum possible Euclidean distance between the discrete patterns.

The choice of parameters A, k and s in Eq. 8 is very important: from one side, they define how well the pseudo-histogram approximates the true frequency distribution; from the other side, they are responsible for smoothing and, consequently, for the convergence properties. Figure 3 reflects how different values of k, s with fixed $A = 100$ influence the shape of the pattern similarity function (Eq. 8). Our empirical conclusion is that values $A = 100, k = 2, s = 2$ are optimal. Compare them (Fig. 3) with the extreme case $A = 100, k = 1, s = 2$ where the majority of patterns have a close-to-zero contribution.

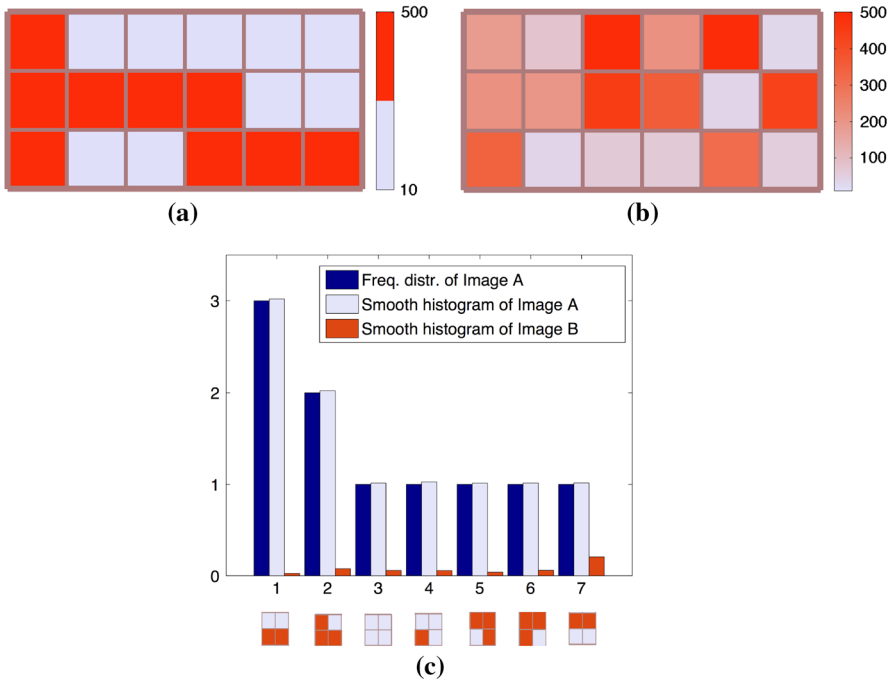

Fig. 2 Pattern statistics represented by frequency distribution and smooth histograms

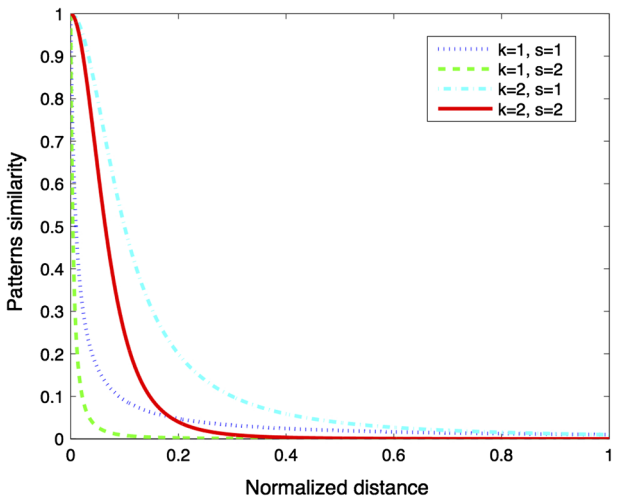

Fig. 3 Pattern similarity function

Comparing the pseudo-histograms quantitatively, we are able to understand how well the multiple-point statistics of the training image is reproduced in the model image. We introduce the following dissimilarity function

$$f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI}) = \frac{1}{2} \sum_{i=1}^{N^{\mathbf{TI},\mathbf{un}}} \frac{(H_i^{\mathrm{d},\mathbf{m}} - H_i^{\mathrm{d},\mathbf{TI}})^2}{H_i^{\mathrm{d},\mathbf{TI}}}. \tag{11}$$

Essentially, it is a weighted ℓ^2-norm, where the role of the weight parameter is played by the smooth histogram of the training image. The suggested measure enhances influence of the less frequent patterns of the training image and improves reproduction of the features. If the number of patterns in the training image $N^{\mathbf{TI}}$ differs from the number of patterns in the model $N^{\mathbf{m}}$, we multiply $H_i^{\mathrm{d},\mathbf{TI}}$ by $r = N^{\mathbf{m}}/N^{\mathbf{TI}}$. Algorithm 1 summarizes the main steps for constructing the dissimilarity function $f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI})$.

Algorithm 1: Construction of the multiple-point statistics dissimilarity function

Input: Training image **TI**, model image **m**, template **T**
Output: $f^d(\mathbf{m}, \mathbf{TI})$
Collect patterns $pat^{\mathbf{TI}}$ and $pat^{\mathbf{m}}$ of **TI** and **m**
Collect unique patterns $pat^{\mathbf{TI},\mathbf{un}}$ of **TI**
Construct the pseudo-histogram of the model image:
for $j = 1 : N^{\mathbf{TI},un}$ **do**
  $H_j^{d,\mathbf{m}} = \sum_{i=1}^{N^{\mathbf{m}}} \frac{1}{(1+A\,||pat_i^{\mathbf{m}} - pat_j^{\mathbf{TI},\mathrm{un}}||_2^k)^s}$
end
Construct the pseudo-histogram of the training image:
for $j = 1 : N^{\mathbf{TI},un}$ **do**
  $H_j^{d,\mathbf{TI}} = \sum_{i=1}^{N^{\mathbf{TI}}} \frac{1}{(1+A\,||pat_i^{\mathbf{TI}} - pat_j^{\mathbf{TI},\mathrm{un}}||_2^k)^s}$
end
Compute the dissimilarity function $f^d(\mathbf{m}, \mathbf{TI})$ (Eq. 10)

2.3 Relation of the Dissimilarity Measure to Prior Probability

In this section, we show how the value of prior probability density $\rho(\mathbf{m})$ can be estimated and how it is related to the dissimilarity function (Eq. 11). We define the prior probability of the model parameters through their marginal probabilities, which can be estimated by constructing the frequency distribution. In other words, by maximizing the probability of the histogram to be a realization of the process that generated the histogram of the training image, we maximize the probability of the image to share the same multiple-point statistics as the training image. Our idea consists in representing an image as an outcome of some multinomial experiment (see also Cordua et al. 2012). Consider two categorical images: training and test. Assume that a pattern in the test image is a multiple-point event that leads to the success for exactly one of the K categories, where each category has a fixed probability of success p_i. By definition, each element H_i in the frequency distribution $\mathbf{H}$ indicates the number of times the ith category has appeared in N trials (number of patterns observed in the test image). Then the vector $\mathbf{H} = (H_1, \ldots, H_K)$ follows the multinomial distribution with parameters N and $\mathbf{p}$, where $\mathbf{p} = (p_1, \ldots, p_K)$

$$\rho(\mathbf{m}) = P(\mathbf{H}) = \frac{N!}{H_1! \cdots H_K!} p_1^{H_1} \cdots p_K^{H_K}. \tag{12}$$

We assume that the histogram of the training image defines the theoretical distribution underlying the multinomial experiment. Then the vector of probabilities $\mathbf{p}$ can be obtained from the frequency distribution of the training image $\mathbf{H}^{\mathbf{TI}}$: by normalizing its entries on the total number of counts we obtain the probabilities of success. In general, the histogram of the training image is very sparse, therefore many categories of patterns will be assigned zero probability. It means that if a test image has a single pattern that is not encountered in the training image, its prior probability from Eq. 12 will be zero. This happens due to the insufficient prior information derived from the training image; it is very likely, however, that many of the non-observed patterns have some non-zero probabilities to exist. This problem is well known in the field of the natural language processing (NLP): small vocabulary can imply zero probabilities of some words to exist. The NLP research community address the challenge with a fundamental technique called "smoothing" (Chen and Goodman 1999). The common idea of smoothing algorithms lies in making prior distributions more uniform by adjusting low probabilites upward and high probabilities downward. Since there is no information about the probabilities of the patterns not encountered in the training image, we assume them to be equal to ε. To make the sum of p_i equal to one, we subtract a small number γ from all non-zero bins of $\mathbf{H}^{\mathbf{TI}}$

$$p_i = \begin{cases} \frac{H_i^{\mathbf{TI}} - \gamma}{N^{\mathbf{TI}}} & H_i^{\mathbf{TI}} > 0 \\ \varepsilon & H_i^{\mathbf{TI}} = 0 \end{cases}, \tag{13}$$

where $\gamma = \varepsilon(K - N^{\mathbf{TI},\text{unique}})N^{\mathbf{TI}}/N^{\mathbf{TI},\text{un}}$.

This simple technique called absolute discounting is one of the many smoothing techniques, however, to define which smoothing methodology is the best for the training-image-based prior is the subject of a separate research and thus we do not address it here. After defining p_i, $P(\mathbf{H})$ can be computed through its logarithm

$$\log(P(\mathbf{H})) = \log\left(\frac{N!}{H_1! \cdots H_K!}\right) + \sum_{i=1}^{K} H_i \log(p_i). \tag{14}$$

Further we apply Stirling's approximation

$$\log(n!) = n \log n - n + O(\log n). \tag{15}$$

Defining $I = \{i : H_i > 0\}$ we obtain

$$\log\left(\frac{N!}{H_1! \cdots H_k!}\right) = \log(N!) - \sum_{i \in I} \log(H_i!) \approx N \log N - N - \sum_{i \in I} (H_i \log(H_i) - H_i) = N \log N - \sum_{i \in I} H_i \log(H_i). \tag{16}$$

and finally

$$\log(P(\mathbf{H})) \approx N \log N + \sum_{i \in I} H_i \log\left(\frac{p_i}{H_i}\right) = \sum_{i \in I} H_i \log\left(\frac{N p_i}{H_i}\right). \quad (17)$$

Having at hand a discrete image, one can compute its relative prior probability using Eq. 17. Moreover, it is also applicable to the result of minimization of Eq. 11, since the algorithm aims at finding an image, whose pixel values are very close to the expected categorical values and therefore its patterns can be considered as a success in the multinomial experiment. The misfit with the prior information can be then written as

$$-\log(P(\mathbf{H})) \approx \sum_{i \in I} H_i \log\left(\frac{H_i}{N p_i}\right). \quad (18)$$

Substituting H_i with $N p_i + \varepsilon_i$ and applying Taylor expansion of the second order one arrives to the chi-square distance divided by two

$$-\log(P(\mathbf{H})) \approx \frac{1}{2} \sum_{i \in I} \frac{(H_i - N p_i)^2}{N p_i}. \quad (19)$$

Notice that Eq. 19 justifies our choice of the dissimilarity function (Eq. 11). Indeed, by minimizing expression 11 we minimize the value defined by Eq. 19 as well. Further, if we denote $\mathbf{h} = \mathbf{H}/N$, Eq. 17 is transformed as

$$\log(P(\mathbf{H})) \approx \sum_{i \in I} N h_i \log\left(\frac{p_i}{h_i}\right) = -\sum_{i \in I} N h_i \log\left(\frac{h_i}{p_i}\right) = -N D_{KL}(h||p), \quad (20)$$

where $D_{KL}(h||p)$ is the Kullback–Leibler divergence, a dissimilarity measure between two probability distributions h and p. In other words, it defines the information lost when the theory (training image) is used to approximate the observations (test image).

2.4 Generating Near-Maximum A Priori Models

Minimizing Eq. 11, we are able to generate near-maximum a priori model, given a starting guess. We solve the following optimization problem

$$\mathbf{m}^{\text{HighPrior}} = \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ f^{\text{d}}(\mathbf{m}, \mathbf{TI}) \right\}. \quad (21)$$

To use an efficient unconstrained optimization framework in case of non-negative model parameters (such as permeability), we apply the logarithmic scaling of the parameters (Gao and Reynolds 2006)

$$x_i = \log\left(\frac{m_i - m^{\text{low}}}{m^{\text{up}} - m_i}\right). \quad (22)$$

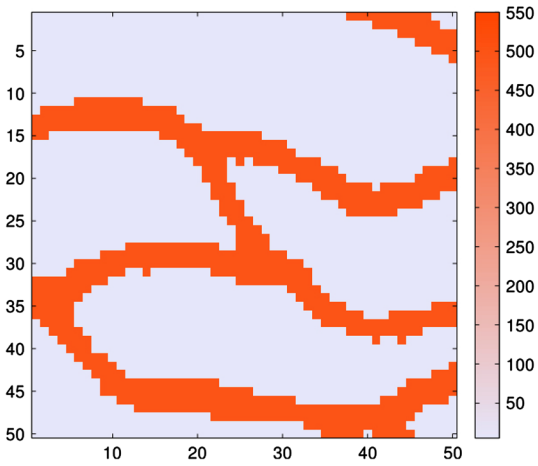

Fig. 4 Training image representing permeability field (mD) used in the numerical examples

Here $i = 1, \ldots, n$, where n is the number of pixels in the test image m, m^{low} and m^{up} are the lower and upper scaling boundaries of the parameters. The log transform does not allow extreme values of the model parameters and makes the algorithm perform in a more robust way. For consistency we transform the training image as well.

Any gradient-based optimization technique can be used for solving Eq. 21, however we used a quasi-Newton method, which was our method of preference when solving inverse problems (Sect. 2.5). It requires only the value of the objective function and its gradient, while the Hessian needed for the search direction is evaluated through approximation (Nocedal and Wright 2006). Appendix A shows how to compute the gradient of Eq. 11 analytically. In this work, we employed the unconstrained implementation of the L-BFGS method (Zhu et al. 1997). Here follows an example. Consider a training image (Fig. 4), which is an upscaled part of a training image proposed by Strebelle (2000). We assume that it represents permeability of an oil reservoir with a values of 500 mD in channels and 10 mD in the background.

To derive the multiple-point statistics, we used a square template of 6×6 pixels [optimal size according to the entropy approach suggested by Honarkhah (2011)]. The training image has 789 unique 6×6 patterns, therefore, the pseudo-histograms (Eq. 11) have 789 bins. Parameters A, k and s (Eq. 8) were set to the empirically optimal values of 100, 2 and 2. Figure 5a shows three starting guesses: one random, and two upscaled smoothed parts of the aforementioned image of Strebelle (2000). Figure 5b shows the solutions after 20 iterations. Finally, Fig. 5c demonstrates the solutions obtained after 100 iterations. Since unconstrained optimization is used, the solutions have few outliers; nevertheless, the logarithmic transformation used in the optimization allows us to regulate the boundaries of pixel values. In this example the minimum possible value is 5 mD, and the maximum is 550 mD. The solutions clearly reproduce features of the training image. The value of the misfit with prior (Eq. 11) is close to 100.0.

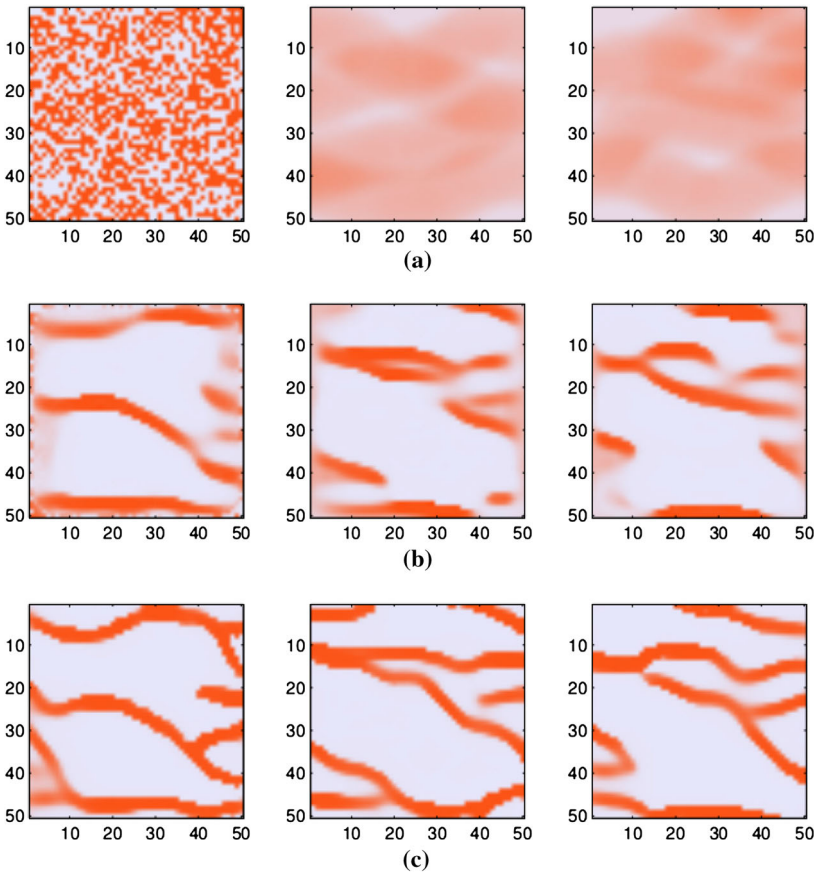

Fig. 5 Generating near-maximum a priori models

2.5 Solving Inverse Problems

It would be tempting to find a high posterior model by minimizing the objective function

$$O(\mathbf{m}) = \frac{1}{2}||\mathbf{d^{obs}} - g(\mathbf{m})||^2_{C_d} + f^{\mathrm{d}}(\mathbf{m}). \tag{23}$$

However, the two terms in this objective function have different dimensions and scales; this may lead to inconsistency in optimization. We overcome these difficulties transforming the current objective terms into dimensionless ones. For the current implementation we used the following expression (Osyczka 1978)

$$F_i^{\mathrm{trans}}(x) = \frac{F_i(x) - F_i^*}{F_i^*}. \tag{24}$$

Here $F_i(x)$ is the ith function to transform, and F_i^* is the target (desired) value of the objective function value. We denote the target value of the data misfit term as u^*, and

Table 2 Reservoir model parameters

Model size	50×50 cells
Cell size	10×10 m
Initial water saturation	0.0
Porosity	0.3 (constant everywhere)

from Oliver et al. (2008) expect $u^* \approx N/2$, where N is the number of observations. The target value of the prior misfit f^* is non-zero, since the training image and images statistically similar to it have slightly different histograms. However the order of magnitude of f^*, which corresponds to the well reproduced features of the training image, is the same and can be found empirically. It can be estimated by finding, for instance, the value of $f^{\mathrm{d}}(m^*)$, where m^* is an image honoring multiple-point statistics of the training image. Alternatively, the order of f^* can be found solving Eq. 21 for some starting model. One of the easiest ways to combine objective functions into a single function is to use the weighted exponential sum (Marler and Arora 2004). We put equal weights on two misfit terms and the exponent equal to 2. This leads to the final expression for the objective function

$$O^*(\mathbf{m}) = \left(\frac{\frac{1}{2}||\mathbf{d^{obs}} - g(\mathbf{m})||^2_{C_d} - u^*}{u^*} \right)^2 + \left(\frac{f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI}) - f^*}{f^*} \right)^2. \qquad (25)$$

Notice that the term with the largest difference between its current and target values gets higher priority. Essentially, u^* and f^* play the role of weights, and the exact values do not need to be known, only the order of magnitude is important. In practice, target values can be set below the desired values to provide faster convergence.

Similarly to Sect. 2.4, we apply the logarithmic transformation (Eq. 22) to the model and to the training image. For solving (25), we suggest using quasi-Newton methods that are known to be efficient for history matching problems (Oliver et al. 2008). The gradient of the data misfit term is calculated by an adjoint method implemented in the reservoir simulator Eclipse (Schlumberger GeoQuest 2009). The gradient of the prior term is computed analytically (Appendix A). The algorithm is stopped when the values of the objective terms in the optimization problem (25) approach their target values. The computational efficiency of the algorithm decreases with increase of the number of categories in the training image and/or the template size, since a larger number of Euclidean distances is to be calculated.

3 History Matching Example

We perform history matching on a two-dimensional synthetic oil reservoir, aiming at estimating its permeability field. All other parameters, such as porosity, relative permeabilities and initial saturation are assumed to be known. To investigate non-uniqueness of the solution we solve Eq. 25 for a set of starting models. Table 2 lists some parameters of the reservoir model.

Figure 6 shows the true permeability field that features sand channels of 500 mD and background shale of 10 mD; 13 injectors are marked by triangles, and 13 producers by

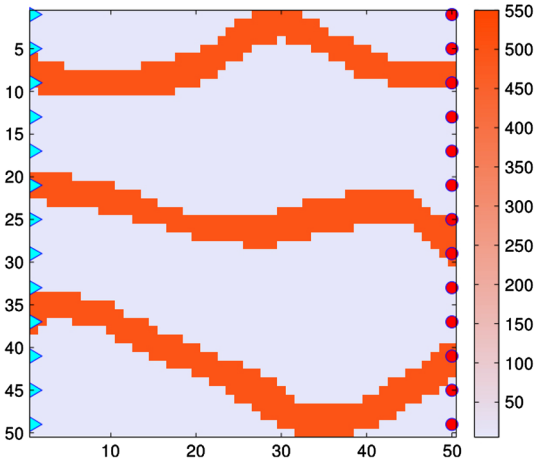

Fig. 6 True model of permeability (mD) with injection and production wells (*triangles* and *circles*, respectively)

circles, respectively. All wells work at the bottom hole pressure control: 300 Barsa for the injectors and 50 Barsa for the producers. Production data are generated by running a forward simulation with the true permeability model and adding 5 % of Gaussian noise. Physics of the flow (steady two-phase immiscible displacement) allows us to use few observations and not to lose in history matching accuracy. We choose just two measurements (at 100 and 200 days) per well, 52 measurements in total (we measure water rate in injectors and oil rate in producers). This approach results in faster performance, since much less time is required to compute sensitivities. However, we show the full history to assure the quality of history matching.

Prior information is given by the training image in Fig. 4. We use the same parameters as in Sect. 2.4 to derive multiple-point statistics and construct the objective function. The ensemble of ten starting guesses (Fig. 7a) is presented by randomly chosen parts of a smoothed and upscaled version of the training image proposed by Strebelle (2000). Solving Eq. 25, we set target values of u^* and f^* at 10.0 and 25.0 to assure the convergence of the algorithm to the desired values of the misfits. For the data misfit we expect a value close to $N/2$ where N is the number of measurements (Oliver et al. 2008) and for the prior close to 10^2. On average the algorithm converges in 100 iterations; its performance depends on the closeness of the initial guess to the solution. Figure 7b demonstrates the transformation of the models after 50 iterations: most of the original channels are blurred and new ones are being constructed. Figure 7c shows models at the 150th iteration. The algorithm successfully reproduces high-contrast channels featuring the expected continuity and width. Naturally, since the data sensitivity decreases with increasing distance from a well, the location of channels is very well defined on the sides of the model, in the vicinity of wells, while in the middle we observe some deviation from the true model. This example clearly demonstrates the consequences of the underdetermined inverse problem: existence of many solutions all satisfying the available information. Figure 8a shows history matching for the first solution: injection rates of the first four injectors and production rates of the first four producers (counting from top). Convergence plot for the prior and the data misfit is

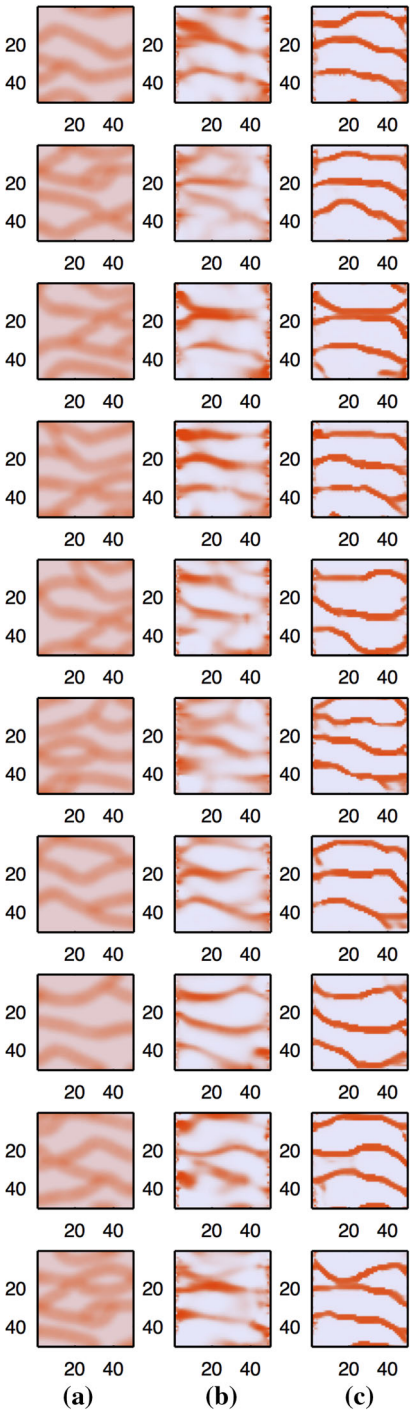

Fig. 7 **a** Starting models, **b** models after 50 iterations, **c** models after 150 iterations

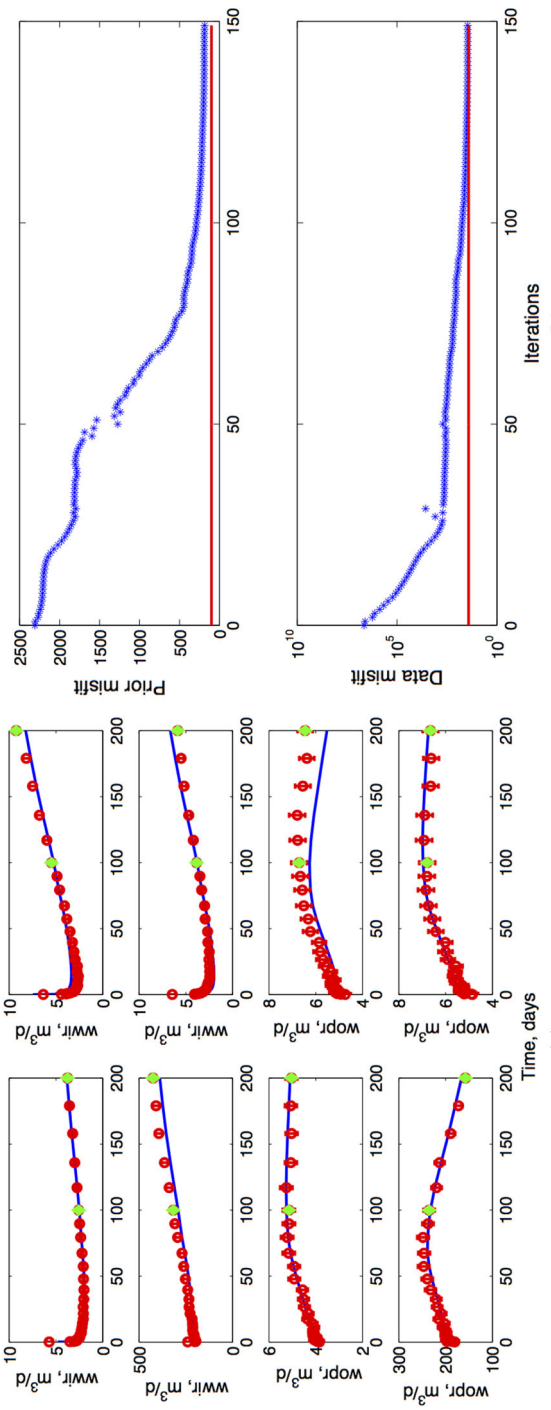

Fig. 8 **a** History matching for the first solution (*green diamonds* observations used in the optimization, *red circles* full history, *blue line* simulated response), **b** convergence of the misfits with the prior and the data

Table 3 Posterior ranking of the solutions

Model N	$\log(\rho(\mathbf{m})) - \frac{1}{2}\|\|g(\mathbf{m}) - \mathbf{d}^{\mathbf{obs}}\|\|^2_{C_D}$
1	−8122.0324
2	−8134.6031
3	−10383.1467
4	−7860.2211
5	−6568.7915
6	−8900.5525
7	−9781.7611
8	−7107.3847
9	−6734.4299
10	−7608.2761
True model	−7713.9272

shown in Fig. 8b (notice log scale for the data misfit term). Red lines mark the desired values of the misfits.

Finally, we are able to distinguish among the solutions (Fig. 7c) by calculating their relative posterior probabilities derived from Eqs. 2 and 3

$$\log(\sigma(\mathbf{m})/(kc_1)) = \log(\rho(\mathbf{m})) - \frac{1}{2}||g(\mathbf{m}) - \mathbf{d}^{\mathbf{obs}}||^2_{C_D} \tag{26}$$

where $\log(\rho(\mathbf{m}))$ is defined by Eq. 17. We chose $\gamma = 0.1$ (Eq. 13). Table 3 lists the results (enumeration of the models starts from top).

For comparison, in the last row, we give the value calculated for the true model (Fig. 6). We can conclude that models 5, 8 and 9 are the most preferable within this ensemble, while model 3 is the most inferior.

4 Conclusions

We presented an efficient method for solving the history matching problem employing a gradient-based optimization technique that integrates complex a priori information (in the form of a training image). History matching is a severely undetermined inverse problem and existence of multiple solutions is a direct (and unfortunate) consequence of this property. However, production data contain valuable information about rock properties, such as porosity and permeability. Inversion of them is necessary for construction of reservoir models that can be used in prediction. Geological information, if available, can drastically decrease the size of the solution space, hence reducing the non-uniqueness of the solution. One way of applying the methodology is to explore the solution space. Since we are able to start from any smooth model in many cases we can detect solutions that have high posterior values and look very different, due to the fact that they belong to the different islands of high probability. Quantification of the relative posterior probabilities allows us to rank solutions and choose the most reliable ones.

The algorithm needs a starting guess, and, clearly as in any gradient-based optimization, the convergence properties depend on it. In the history matching problem, the choice of the starting guess is particularly important. The sensitivity of the production data with respect to the rock properties decreases non-linearly with the distance from wells. Therefore, it is hard to invert for model parameters in the areas with poor well coverage. The situation can be greatly simplified if one would integrate seismic data, or at least, would use the results of the seismic inversion as the starting guesses. This is a topic of our future research.

Acknowledgments The present work was sponsored by the Danish Council for Independent Research-Technology and Production Sciences (FTP Grant No. 274-09-0332) and DONG Energy.

Appendix A: Computing Gradient of the Dissimilarity Function

In order to perform the gradient-based optimization of Eqs. 21 and 25 the derivatives of the dissimilarity function $f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI})$ (Eq. 11) with respect to model parameters have to be computed. Below we show how to compute this gradient analytically.

By definition, $\nabla f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI}) = \left[\frac{\partial f^{\mathrm{d}}}{\partial m_1}, \ldots, \frac{\partial f^{d}}{\partial m_n}\right]^T$ where $\mathbf{m} \in \mathbf{R}^n$. From Eq. 11 it reads

$$\nabla f^{\mathrm{d}}(\mathbf{m}, \mathbf{TI}) = \begin{bmatrix} \frac{\partial H_1^{\mathrm{d},\mathbf{m}}}{\partial m_1} & \frac{\partial H_2^{\mathrm{d},\mathbf{m}}}{\partial m_1} & \cdots & \frac{\partial H_{N^{\mathbf{TI},\mathrm{un}}}^{\mathrm{d},\mathbf{m}}}{\partial m_1} \\ \frac{\partial H_1^{\mathrm{d},\mathbf{m}}}{\partial m_2} & \frac{\partial H_2^{\mathrm{d},\mathbf{m}}}{\partial m_2} & \cdots & \frac{\partial H_{N^{\mathbf{TI},\mathrm{un}}}^{\mathrm{d},\mathbf{m}}}{\partial m_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial H_1^{\mathrm{d},\mathbf{m}}}{\partial m_n} & \frac{\partial H_2^{\mathrm{d},\mathbf{m}}}{\partial m_n} & \cdots & \frac{\partial H_{N^{\mathbf{TI},\mathrm{un}}}^{\mathrm{d},\mathbf{m}}}{\partial m_n} \end{bmatrix} \begin{bmatrix} \frac{H_1^{\mathrm{d},\mathbf{m}} - H_1^{\mathrm{d},\mathbf{TI}}}{H_1^{\mathrm{d},\mathbf{TI}}} \\ \frac{H_2^{\mathrm{d},\mathbf{m}} - H_2^{\mathrm{d},\mathbf{TI}}}{H_2^{\mathrm{d},\mathbf{TI}}} \\ \vdots \\ \frac{H_{N^{\mathbf{TI},\mathrm{un}}}^{\mathrm{d},\mathbf{m}} - H_{N^{\mathbf{TI},\mathrm{un}}}^{\mathrm{d},\mathbf{TI}}}{H_{N^{\mathbf{TI},\mathrm{un}}}^{\mathrm{d},\mathbf{TI}}} \end{bmatrix}. \tag{27}$$

From Eqs. 10 and 8 it follows

$$\frac{\partial H_j^{\mathrm{d},\mathbf{m}}}{\partial m_z} = \sum_{i=1}^{N^{\mathbf{m}}} \frac{\partial c_{ij}}{\partial m_z} = \sum_{i=1}^{N^{\mathbf{m}}} -Aks(1 + At_{ij}^k)^{(-s-1)} t_{ij}^{k-1} \frac{\partial t_{ij}}{\partial m_z}, \tag{28}$$

where $i = 1, \ldots, N^{\mathbf{m}}$, $j = 1, \ldots, N^{\mathbf{TI},\mathrm{un}}$ and $z = 1, \ldots, n$.

Notice, that $\frac{\partial t_{ij}}{\partial m_z} = 0$ if $m_z \notin pat_i^{\mathbf{m}}$. Otherwise, if $pat_i^{\mathbf{m}} = [v_{i,1} \ldots v_{i,N}]^T$, and $pat_j^{\mathbf{TI},\mathrm{un}} = [u_{j,1} \cdots u_{j,N}]^T$, where N is the number of pixels in the pattern, we get

$$t_{ij} = ||\mathrm{pat}_i^{\mathbf{m}} - \mathrm{pat}_j^{\mathbf{TI}}||_2 = \sqrt{(v_{i,1} - u_{j,1})^2 + \ldots, + (v_{i,N} - u_{j,N})^2}, \tag{29}$$

and, therefore

$$\frac{\partial t_{ij}}{\partial m_z} = \frac{v_{i,s} - u_{j,s}}{||\mathrm{pat}_i^{\mathbf{m}} - \mathrm{pat}_j^{\mathbf{TI}}||_2}, \tag{30}$$

where $v_{i,s} = m_z$.

References

Caers J (2003) History matching under training-image-based geological model constraints. SPE J 8:218–226
Caers J, Hoffman T (2006) The probability perturbation method: a new look at bayesian inverse modeling. Math Geol 38:81–100
Chen SF, Goodman J (1999) An empirical study of smoothing techniques for language modeling. Comput Speech Lang 13:359–394
Cordua KS, Hansen TM, Lange K, Frydendall J, Mosegaard K (2012a) Improving multiple-point-based a priori models for inverse problems by combining sequential simulation with the frequency matching method. Paper presented at 82th annual meeting for the society of exploration geophysicists (SEG 2012), Las Vegas, NE, United States
Cordua KS, Hansen TM, Mosegaard K (2012b) Monte carlo full waveform inversion of crosshole gpr data using multiple-point geostatistical a priori information. Geophysics 77:H19–H31
Gao G, Reynolds AC (2006) An improved implementation of the lbfgs algorithm for automatic history matching. SPE J 11(1):5–17
Guardiano F, Srivastava RM (1993) Multivariate geostatistics: beyond bivariate moments. In: Soares A (ed), vol 1, Geostatistics Troia, Kluwer Academic
Hansen TM, Mosegaard K, Cordua KS (2009) Reducing complexity of inverse problems using geostatistical priors. In: Proceedings of IAMG 09
Hansen TM, Cordua KS, Mosegaard K (2012) Inverse problems with non-trivial priors: efficient solution through sequential gibbs sampling. Comput Geosci 16:593–611
Honarkhah M (2011) Stochastic simulation of patterns using distance-based pattern modeling. PhD thesis, Stanford University
Jafarpour B, Khodabakhshi M (2011) A probability conditioning method (PCM) for nonlinear flow data integration into multipoint statistical facies simulation. Math Geosci 43:133–164
Lange K, Frydendall J, Cordua KS, Hansen TM, Melnikova Y, Mosegaard K (2012) A frequency matching method: solving inverse problems by use of geologically realistic prior information. Math Geosci 44: 783–803
Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidisc Optim 26:369–395
Mosegaard K, Tarantola A (1995) Monte carlo sampling of solutions to inverse problems. J Geophys Res 100:12431–12447
Nocedal J, Wright SJ (2006) Numerical optimization. Springer, Berlin
Oliver DS, Reynolds AC, Liu N (2008) Petroleum reservoir characterization and history matching. Cambridge University Press, New York
Osyczka A (1978) An approach to multicriterion optimization problems for engineering design. Comput Meth Appl Mech Eng 15:309–333
Sarma P, Durlofsky LJ, Aziz K (2008) Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. Math Geosci 40:3–32
Schlumberger GeoQuest (2009) ECLIPSE reservoir simulator. Technical description
Strebelle S (2000) Sequential simulation drawing structures from training images. PhD thesis, Stanford University
Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. Math Geol 34(1):1–20
Tarantola A (2005) Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics, Philadelphia
Zhu C, Byrd RH, Lu P, Nocedal J (1997) L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS) 23(4):550–560